**E-Com Application (Coding Challenge)**

**J211-Logeshwari K**

create database assignment;

use assignment;

//customers table

create table customers

(

customer\_id int primary key,

firstname varchar(40),

lastname varchar(40),

email varchar(40),

password varchar(40)

);

insert into customers (customer\_id, firstname, lastname, email, password) values

(1, 'John', 'Doe', 'johndoe@example.com', '123 Main St, City'),

(2, 'Jane' ,'Smith', 'janesmith@example.com', '456 Elm St, Town'),

(3, 'Robert', 'Johnson', 'robert@example.com', '789 Oak St, Village'),

(4, 'Sarah', 'Brown', 'sarah@example.com', '101 Pine St, Suburb'),

(5, 'David', 'Lee', 'david@example.com', '234 Cedar St, District'),

(6, 'Laura', 'Hall', 'laura@example.com', '567 Birch St, County'),

(7, 'Michael', 'Davis', 'michael@example.com', '890 Maple St, State'),

(8, 'Emma', 'Wilson', 'emma@example.com', '321 Redwood St, Country'),

(9, 'William', 'Taylor', 'william@example.com', '432 Spruce St, Province'),

(10, 'Olivia', 'Adams', 'olivia@example.com', '765 Fir St, Territory');

//products table

create table products

(

product\_id int primary key,

name varchar(40),

description varchar(50),

price float,

stockQuantity int

);

insert into products (product\_id, name, description, price, stockQuantity) VALUES

(1, 'Laptop', 'High-performance laptop', 800.00, 10),

(2, 'Smartphone', 'Latest smartphone', 600.00, 15),

(3, 'Tablet', 'Portable tablet', 300.00, 20),

(4, 'Headphones', 'Noise-canceling', 150.00, 30),

(5, 'TV', '4K Smart TV', 900.00, 5),

(6, 'Coffee Maker', 'Automatic coffee maker', 50.00, 25),

(7, 'Refrigerator', 'Energy-efficient', 700.00, 10),

(8, 'Microwave Oven', 'Countertop microwave', 80.00, 15),

(9, 'Blender', 'High-speed blender', 70.00, 20),

(10, 'Vacuum Cleaner', 'Bagless vacuum cleaner', 120.00, 10);

//cart table

create table cart

(

cart\_id int primary key,

customer\_id int,

product\_id int,

quantity int,

foreign key (customer\_id) references customers(customer\_id),

foreign key (product\_id) references products(product\_id)

);

insert into cart (cart\_id, customer\_id, product\_id, quantity) values

(1, 1, 1, 2),

(2, 1, 3, 1),

(3, 2, 2, 3),

(4, 3, 4, 4),

(5, 3, 5, 2),

(6, 4, 6, 1),

(7, 5, 1, 1),

(8, 6, 10, 2),

(9, 6, 9, 3),

(10, 7, 7, 2);

//order table

create table orders

(

order\_id int primary key,

customer\_id int,

order\_date date,

total\_price float,

foreign key (customer\_id) references customers(customer\_id)

);

insert into orders (order\_id, customer\_id, order\_date, total\_price) values

(1, 1, '2023-01-05', 1200.00),

(2, 2, '2023-02-10', 900.00),

(3, 3, '2023-03-15', 300.00),

(4, 4, '2023-04-20', 150.00),

(5, 5, '2023-05-25', 1800.00),

(6, 6, '2023-06-30', 400.00),

(7, 7, '2023-07-05', 700.00),

(8, 8, '2023-08-10', 160.00),

(9, 9, '2023-09-15', 140.00),

(10, 10, '2023-10-20', 1400.00);

//order\_items table

create table order\_items

(

order\_item\_id int primary key,

order\_id int,

product\_id int,

quantity int,

itemAmount float,

foreign key (order\_id) references orders(order\_id),

foreign key (product\_id) references products(product\_id)

);

insert into order\_items (order\_item\_id, order\_id, product\_id, quantity, itemAmount) values

(1, 1, 1, 2, 1600.00),

(2, 1, 3, 1, 300.00),

(3, 2, 2, 3, 1800.00),

(4, 3, 5, 2, 1800.00),

(5, 4, 4, 4, 600.00),

(6, 4, 6, 1, 50.00),

(7, 5, 1, 1, 800.00),

(8, 5, 2, 2, 1200.00),

(9, 6, 10, 2, 240.00),

(10, 6, 9, 3, 210.00);

drop table customers;

drop table products;

drop table cart;

drop table orders;

drop table order\_items;

select \* from customers;

select \* from products;

select \* from cart;

select \* from orders;

select \* from order\_items;

-- Query1

update products

set price=800.00

where product\_id=7 and name='Refrigerator';

-- Query2

delete from cart

where customer\_id=6;

-- Query3

select name from products

where price<100;

-- Query4

select name from products

where stockQuantity>5;

-- Query5

select \* from orders

where total\_price between 500 and 1000;

-- Query6

select name from products

where name like '%r';

-- Query7

select p.name from cart c

inner join products p

on c.product\_id=p.product\_id

where c.customer\_id=5;

-- Query8

select c.firstname,c.lastname from customers c

inner join orders o

on c.customer\_id=o.customer\_id

where o.order\_date between '2023-01-01' and '2023-12-31';

-- Query9

select min(stockQuantity) from products;

-- Query10

select o.customer\_id,sum(oi.itemAmount) from orders o

inner join order\_items oi

on o.order\_id=oi.order\_id

group by o.customer\_id;

-- Query11

select customer\_id,avg(total\_price) as Average\_amount from orders

group by customer\_id;

-- Query12

select o.customer\_id, count(\*) as number\_of\_orders from orders o

inner join order\_items oi

on o.order\_id=oi.order\_id

group by oi.order\_id;

-- Query13

select o.customer\_id, max(itemAmount) as maximum\_amount from orders o

inner join order\_items oi

on o.order\_id=oi.order\_id

group by oi.order\_id;

-- Query14

select o.customer\_id, sum(itemAmount) as sum\_amount from orders o

inner join order\_items oi

on o.order\_id=oi.order\_id

group by oi.order\_id

having sum(itemAmount)>=1000;

-- Query15

insert into cart (cart\_id, customer\_id, product\_id, quantity) values

(8, 6, 10, 2),

(9, 6, 9, 3);

select name from products

where product\_id not in (select product\_id from cart);

-- Query16

select customer\_id from orders

where order\_id not in (select order\_id from order\_items);

-- Query17

select product\_id,(sum(itemAmount)/(select sum(itemAmount) from order\_items)) \* 100 as revenue\_percentage

from order\_items

group by product\_id;

-- Query18

select name from products

where stockQuantity = (select min(stockQuantity) from products);

-- Query19

select customer\_id from orders

where order\_id = (select order\_id from order\_items group by order\_id order by sum(itemAmount) desc limit 1);